TEMA 3: IMPLEMENTACIÓN DE CIRCUITOS COMBINACIONALES CON PUERTAS LÓGICAS.

3.1. Representación de funciones: mapas de Karnaugh de hasta 5 variables.

El Mapa de Karnaugh es una representación gráfica de una función booleana. Los minitérminos adjuntos (vecinos) sólo se diferencian en una variable. Se consideran vecinos también los extremos. La numeración de las filas y columnas es un código GRAY.

1) Para dos variables:

A \ B	0	1
0	0	1
1	1	1

A	B	F	
0	0	0	lugar 0
0	1	1	lugar 1
1	0	1	lugar 2
1	1	1	lugar 3

2) Para tres variables:

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	1	0

A	B	C	F	
0	0	0	1	lugar 0
0	0	1	0	lugar 1
0	1	0	1	lugar 2
0	1	1	0	lugar 3
1	0	0	1	lugar 4
1	0	1	0	lugar 5
1	1	0	0	lugar 6
1	1	1	1	lugar 7

3) Para cuatro variables:

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	1	1	0
11	0	1	1	1
10	1	1	0	1

A	B	C	D	F	
0	0	0	0	1	lugar 0
0	0	0	1	0	lugar 1
0	0	1	0	1	lugar 2
0	0	1	1	0	lugar 3
0	1	0	0	1	lugar 4
0	1	0	1	1	lugar 5
0	1	1	0	0	lugar 6
0	1	1	1	1	lugar 7
1	0	0	0	1	lugar 8
1	0	0	1	1	lugar 9
1	0	1	0	1	lugar 10
1	0	1	1	0	lugar 11
1	1	0	0	0	lugar 12
1	1	0	1	1	lugar 13
1	1	1	0	1	lugar 14
1	1	1	1	1	lugar 15

4) Para cinco variables:

AB \ CDE	000	001	011	010	110	111	101	100
00	1	0	0	1	0	0	0	1
01	1	1	1	0	0	1	1	0
11	0	1	1	1	1	0	1	0
10	1	1	0	1	1	1	1	0

Numeración de los lugares en los MK

A \ B	0	1
0	0	1
1	2	3

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

3.2. Simplificación de funciones mediante Mapas de Karnaugh.

1) Sea, por ejemplo, el mapa adjunto:

x \ yz	00	01	11	10
0	0	0	1	1
1	0	1	1	0

La función es la suma de los minitérminos 2, 3, 5 y 7. Es decir,

$$F(x,y,z) = \overline{x}\cdot y\cdot \overline{z} + \overline{x}\cdot y\cdot z + x\cdot \overline{y}\cdot z + x\cdot y\cdot z$$

Agrupando los minitérminos 2 con 3 y los minitérminos 5 con 7 y sacando factor común, queda:

$$F(x,y,z) = \overline{x}\cdot y\cdot(\overline{z} + z) + x\cdot z\cdot(\overline{y} + y) = \overline{x}\cdot y + x\cdot z$$

2) Sea, por ejemplo, el mapa adjunto:

x \ yz	00	01	11	10
0	0	0	0	1
1	0	0	1	1

La función es la suma de los minitérminos 2, 6 y 7. Es decir,

$$F(x,y,z)=\overline{x}\cdot y\cdot \overline{z}+x\cdot y\cdot \overline{z}+x\cdot y\cdot z$$

Desdoblando el minitérmino 6 y agrupando los minitérminos 2 con 6 y los minitérminos 6 con 7 y sacando factor común, queda:

$$F(x,y,z)=\overline{x}\cdot y\cdot \overline{z}+x\cdot y\cdot \overline{z}+x\cdot y\cdot \overline{z}+x\cdot y\cdot z=$$
$$=y\cdot \overline{z}\cdot(\overline{x}+x)+x\cdot y\cdot(\overline{z}+z)=y\cdot \overline{z}+x\cdot y$$

3) Sea, por ejemplo, el mapa adjunto:

x \ yz	00	01	11	10
0	0	0	1	1
1	0	0	1	1

La función es la suma de los minitérminos 2, 3, 6 y 7. Es decir,

$$F(x,y,z)=\overline{x}\cdot y\cdot \overline{z}+\overline{x}\cdot y\cdot z+x\cdot y\cdot \overline{z}+x\cdot y\cdot z$$

Agrupando los minitérminos 2 con 3 y 6 con 7 y sacando factor común, queda:

$$F(x,y,z)=\overline{x}\cdot y\cdot(\overline{z}+z)+x\cdot y\cdot(\overline{z}+z)=\overline{x}\cdot y+x\cdot y=(\overline{x}+x)\cdot y=y$$

¿Cómo realizar estas operaciones de forma sistemática?

1) Realizar agrupaciones de 1's con sus vecinos lo mayor posible pero siempre en cantidades potencias de 2.

2) No dejar ningún 1 sin agrupar. Puede ocurrir que un 1 pertenezca a más de una agrupación. No se pueden coger agrupaciones dentro de agrupaciones.

3) Por cada agrupación de 1's resulta un producto de variables. Cuanto más 1's se agrupen, más sencilla resultará la expresión de esa agrupación. En MK de 5 variables, las agrupaciones que tomen 1's de las dos porciones deben ser simétricas respecto al eje central.

4) En cada agrupación, cada una de las variables puede aparecer en alguno de los siguientes casos:

a) Si siempre vale 1 -----> Se pone afirmada.
b) Si siempre vale 0 -----> Se pone negada.
c) Si cambia de valor (50% de los casos un valor y el otro 50% otro valor)-----> No se pone.

5) La expresión de la función booleana será la suma lógica de todos los productos que hayan salido.

Ejemplos de 2 variables:

A \ B	0	1
0	0	1
1	0	1

$F = B$

A \ B	0	1
0	0	1
1	1	0

$F = \overline{A}B + A\overline{B}$

A \ B	0	1
0	1	1
1	1	0

$F = \overline{A} + \overline{B}$

Ejemplos de 3 variables:

A \ BC	00	01	11	10
0	0	1	1	0
1	1	1	0	0

$F = A\overline{B} + \overline{A}C$

A \ BC	00	01	11	10
0	1	0	0	1
1	1	1	1	0

$F = AC + \overline{B}\overline{C} + \overline{A}\overline{C}$

Ejemplos de 4 variables:

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$F = BD + \overline{A}\overline{B}C$

AB \ CD	00	01	11	10
00	1	1	0	1
01	0	1	1	1
11	1	1	1	0
10	1	0	0	0

$F = BD + A\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C} + \overline{A}C\overline{D}$

Ejemplo de 5 variables:

AB \ CDE	000	001	011	010	110	111	101	100
00	1	1	1	1	1	0	0	1
01	1	1	1	0	0	0	0	1
11	1	1	1	0	0	1	0	1
10	1	0	1	1	1	1	0	1

$F(A,B,C,D,E) = \overline{D}\cdot\overline{E} + \overline{A}\cdot\overline{B}\cdot\overline{C} + B\cdot\overline{C}\cdot E + \overline{B}\cdot D\cdot\overline{E} + A\cdot\overline{C}\cdot D\cdot E + A\cdot C\cdot D\cdot E =$

$= \overline{D}\cdot\overline{E} + \overline{A}\cdot\overline{B}\cdot\overline{C} + B\cdot\overline{C}\cdot E + \overline{B}\cdot D\cdot\overline{E} + A\cdot D\cdot E\cdot(\overline{C}+C) =$

$\overline{D}\cdot\overline{E} + \overline{A}\cdot\overline{B}\cdot\overline{C} + B\cdot\overline{C}\cdot E + \overline{B}\cdot D\cdot\overline{E} + A\cdot D\cdot E$

En caso de que una agrupación de unos abarque las dos mitades, para que sea una agrupación válida se deben repartir los unos al 50% en ambas mitades. Puede ocurrir que dos agrupaciones formen una única agrupación. Para ello deben ser simétricas respecto al eje central del mapa. En este ejemplo ocurre con las señaladas en color rosa.

3.2.1. Simplificación como Suma de Productos y como Productos de Sumas.

A continuación tenemos un ejemplo de una función F de 4 variables que se va a simplificar como suma de productos agrupando unos. En el mapa de la derecha está la función complementaria $\overline{F}$, que también se simplifica como suma de productos agrupando unos.

DC \ BA	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	0	0	0
10	1	1	0	1

$$F = \bar B \bar C + A\bar B \bar D + \bar A \bar C$$

DC \ BA	00	01	11	10
00	0	0	1	0
01	1	0	1	1
11	1	1	1	1
10	0	0	1	0

$$\bar F = AB + CD + \bar A C$$

Si negamos la función $\overline{F}$, quedará F doblemente negada, es decir, la propia F, que ahora tendrá una expresión algebraica dada como producto de sumas.

$$F = \overline{\overline{F}} = \overline{A\cdot B + C\cdot D + \overline{A}\cdot C} = \overline{A\cdot B}\cdot\overline{CD}\cdot\overline{\overline{A}C} = (\overline{A}+\overline{B})\cdot(\overline{C}+\overline{D})\cdot(A+\overline{C})$$

A esta misma expression se habría llegado aplicando el criterio general. Una agrupación de ceros produce una suma (entre paréntesis). Las variables aparecerán según el siguiente criterio:

Las variables que siempre valen 1 aparecen NEGADAS.
Las variables que siempre valen 0 aparecen AFIRMADAS.
Las variables que cambian, desaparecen.

Según este criterio, directamente aplicado al mapa de ceros del mapa de la izquieda, quedaría:

$$F = (\overline{A}+\overline{B})\cdot(\overline{C}+\overline{D})\cdot(A+\overline{C})$$

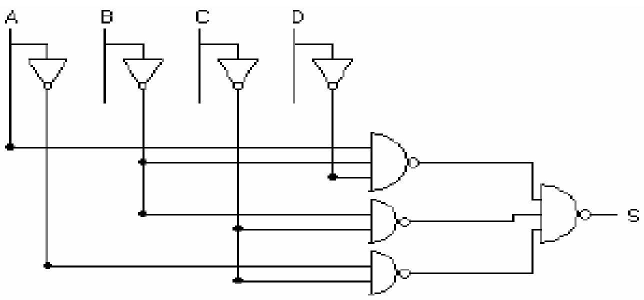
3.2.2. Implementación de funciones sólo con puertas NAND/NOR.

Se basa en hacer $F = \overline{\overline{F}}$ y aplicar una de las leyes de De Morgan al inversor inferior.

Ejemplo: $S = \overline{B}\cdot\overline{C} + A\cdot\overline{B}\cdot\overline{D} + \overline{A}\cdot\overline{C}$

se puede expresar como $S = \overline{\overline{\overline{B}\cdot\overline{C}}\cdot\overline{A\cdot\overline{B}\cdot\overline{D}}\cdot\overline{\overline{A}\cdot\overline{C}}}$

cuya implementación es (SÓLO CON PUERTAS NAND):



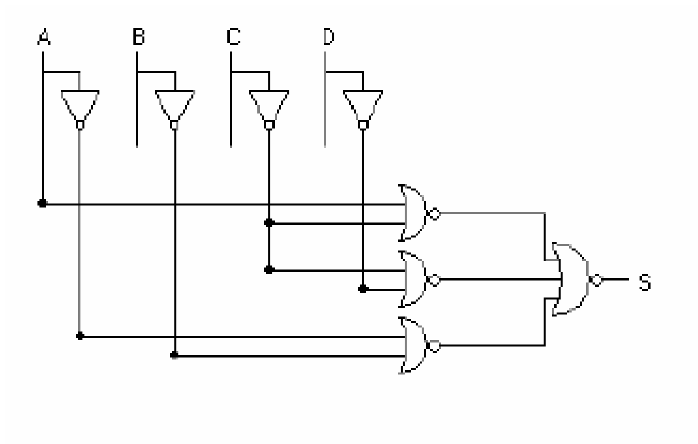
Ejemplo:

$$S = (A + \overline{C}) \cdot (\overline{C} + \overline{D}) \cdot (\overline{A} + \overline{B})$$

se puede expresar como

$$S = \overline{\overline{(A + \overline{C})} + \overline{(\overline{C} + \overline{D})} + \overline{(\overline{A} + \overline{B})}}$$

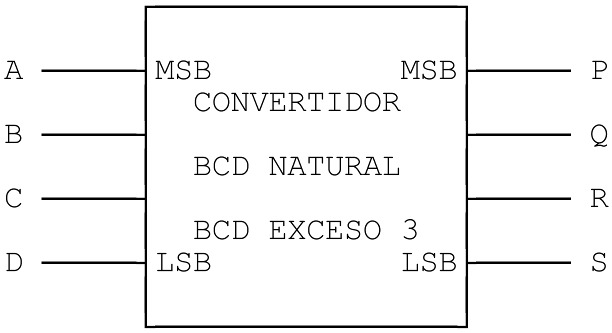
cuya implementación es (SÓLO CON PUERTAS NOR):



3.2.3. Condiciones de “no importa” (don’t care).

Ejemplo: convertidor BCD natural a BCD exceso 3.

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X



Como los códigos 1010 a 1111 no están permitidos en BCD, no tiene sentido plantearse cuáles serán los valores de la función es ese caso, y se ponen “X”, condiciones de “no importa”. A la hora de simplificar, se le dará a las “X” el valor que nos convenga para hacer simplificaciones mas sencillas, en la confianza de que nunca entrará un valor prohibido. Desde el momento que a las condiciones de “no importa” se les asigna un valor, en caso de introducir una entrada prohibida, la salida sería el valor binario que le hayamos asignado a dicha condición de “no importa”.

Mapas y simplificaciones:

P

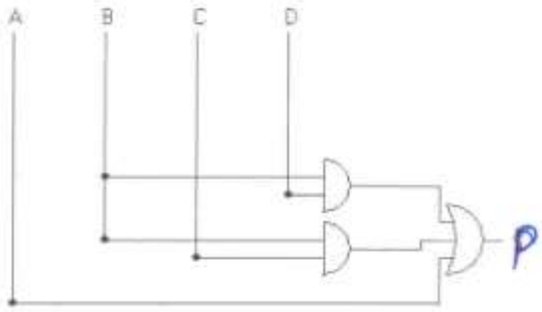
AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

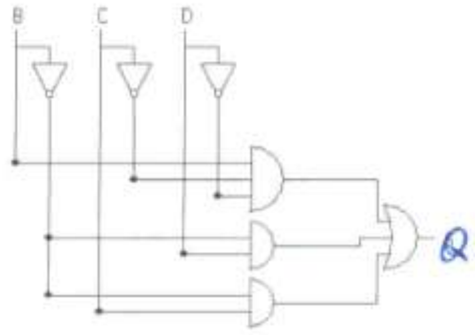
$$P = A + B \cdot C + B \cdot D$$

Q

AB \ CD	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

$$Q = \overline{C} \cdot \overline{D} \cdot B + D \cdot \overline{B} + C \cdot \overline{B}$$





R

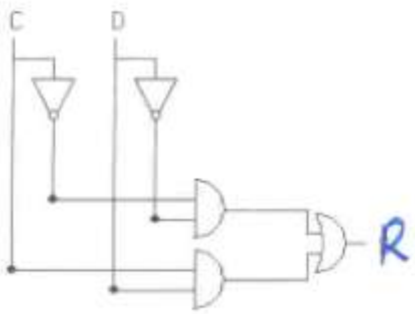
AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

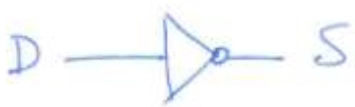
$$R = C \cdot D + \overline{C} \cdot \overline{D}$$

S

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$$S = \overline{D}$$



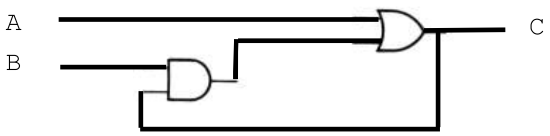


3.3. Definición y Especificación de sistemas combinacionales.

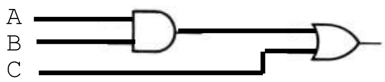
Un sistema combinacional es un circuito lógico cuyas salidas están completamente determinadas en cualquier instante por los valores aplicados a sus variables de entrada.

Son circuitos que no tienen bucles de realimentación, es decir una salida no puede usarse como entrada de una etapa anterior.

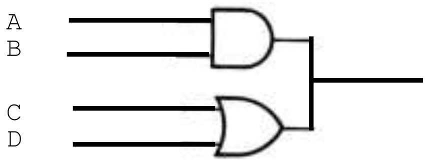
Ejemplo de circuito no combinacional, por tener un bucle de realimentación.



Ejemplo de circuito combinacional



Tampoco se pueden unir entre sí dos salidas porque puede darse un conflicto.



¿A·B ó C+D?

Si A=0 y C=1 se produce un conflicto

Implementación de un sistema combinacional.

La **implementación** de un sistema es su composición en unidades más pequeñas. Ejemplo: Implementar un circuito eléctrico de encendido de una bombilla a partir de una pila, un interruptor y una bombilla.

Especificación de sistemas combinacionales. Ejemplo.

La **especificación** de un sistema combinacional consiste en traducir el enunciado de un problema concreto a variables y funciones booleanas cuya tabla de verdad permita encontrar un circuito lógico que resuelva la situación.

Especificación de alto nivel.

La especificación se dice que es de alto nivel cuando se encuentra una tabla de verdad en la que se reflejan los casos que se pretenden contemplar de manera similar al enunciado del problema. Veamos un ejemplo:

Una máquina expendedora automática proporciona productos con diversos precios: botella de agua 0,50 €, lata de refresco 1,00 €, paquete de galletas 1,50 € y caja de bombones 2,00 €. Sólo admite una moneda de 0,50 €, 1,00 € ó 2,00 € para adquirir el producto y sólo devuelve cambio de 1 moneda, caso de que tuviera que devolver cambio. Habrá casos en los que, al no poder proporcionar el cambio correcto, devolverá la moneda introducida, sin proporcionar el producto.

Realizar la especificación de alto nivel de la máquina:

ENTRADAS		SALIDAS		
Moneda	**Producto pedido**	**¿Suministra producto?**		**Cambio**
0,00 €	Botella de agua	No	*	0,00 €
0,00 €	Lata de refresco	No	*	0,00 €
0,00 €	Paquete de galletas	No	*	0,00 €
0,00 €	Caja de bombones	No	*	0,00 €
0,50 €	Botella de agua	Sí		0,00 €
0,50 €	Lata de refresco	No	*	0,50 €
0,50 €	Paquete de galletas	No	*	0,50 €
0,50 €	Caja de bombones	No	*	0,50 €
1,00 €	Botella de agua	Sí		0,50 €
1,00 €	Lata de refresco	Sí		0,00 €
1,00 €	Paquete de galletas	No	*	1,00 €
1,00 €	Caja de bombones	No	*	1,00 €
2,00 €	Botella de agua	No	**	2,00 €
2,00 €	Lata de refresco	Sí		1,00 €
2,00 €	Paquete de galletas	Sí		0,50 €
2,00 €	Caja de bombones	Sí		0,00 €

MOTIVO DE LA NEGATIVA
* Dinero insuficiente ** No hay cambio en una única moneda

Especificación de bajo nivel.

La especificación se dice que es de bajo nivel cuando los casos posibles se codifican de forma binaria a partir de la especificación de alto nivel. Veamos el mismo ejemplo del apartado anterior especificado a bajo nivel. Vamos a codificar los distintos tipos de monedas con 2 bits, y los distintos tipos de productos también con 2 bits.

ENTRADAS	
Codif. moneda (m1,m0) 00: Ninguna 01: moneda de 0,50 € 10: moneda de 1,00 € 11: moneda de 2,00 €	**Codificación producto (p1,p0)** 00: botella de agua 01: lata de refresco 10: paquete de galletas 11: caja de bombones
SALIDAS	
Codif. devolución (c1,c0) 00: Ninguna 01: moneda de 0,50 € 10: moneda de 1,00 € 11: moneda de 2,00 €	**Codificación suministro (S)** 0: NO da el producto seleccionado 1: SI da el producto seleccionado

Si "traducimos" la especificación de alto nivel mediante la codificación citada, queda la tabla de verdad especificada en bajo nivel, que ya es tratable como funciones y variables booleanas:

ENTRADAS				**SALIDAS**		
Moneda		**Producto pedido**		**¿Suministra producto?**	**Cambio**	
m1	**m0**	**p1**	**p0**	**S**	**c1**	**c0**
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	0
1	1	1	0	1	0	1
1	1	1	1	1	0	0

que es la Tabla de Verdad de 3 funciones booleanas (S, c1 y c0) de 4 variables (m1, m0, p1 y p0) cada una.

El proceso de diseño de un circuito combinacional consiste en:

1) Determinar el número de variables de entrada y de salida necesarias, identificar las variables de entrada, asignarles un nombre y hacer lo mismo con las variables de salida.

2) Deducir la tabla de verdad que define las relaciones entre las variables de entrada y de salida.

3) Simplificar las funciones representadas en la tabla de verdad.

4) Obtener el circuito a partir de las funciones simplificadas.

Si simplificamos aplicando los mapas de Karnaugh resulta:

$$S = m1 \cdot m0 \cdot p0 + m1 \cdot m0 \cdot p1 + m1 \cdot \overline{m0} \cdot \overline{p1} + \overline{m1} \cdot m0 \cdot \overline{p1} \cdot \overline{p0}$$

$$c1 = m1 \cdot m0 \cdot \overline{p1} + m1 \cdot \overline{m0} \cdot p1$$

$$c0 = m1 \cdot \overline{p1} \cdot \overline{p0} + \overline{m1} \cdot m0 \cdot p0 + m0 \cdot p1 \cdot \overline{p0}$$

cuya implementación resulta sencilla a la vista de estas ecuaciones.

EJERCICIOS RESUELTOS

1. Representar el mapa de Karnaugh y simplificar la función booleana

$$F = \sum_4 m(5,6,13) + \sum_4 d(4,7,12,14,15)$$

AB \ CD	00	01	11	10
00	0	0	0	0
01	X	1	X	1
11	X	1	X	X
10	0	0	0	0

F = B

2. Encontrar la tabla de verdad, el mapa de Karnaugh y la expresión booleana más simplificada de una función booleana de 4 variables que tome el valor 1 cuando el número expresado en binario por sus variables sea un número primo mayor que 4, y 0 en el resto de los casos. Expresar la función como suma de productos y como producto de sumas.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

Como suma de productos:

$$F(A,B,C,D) = B \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot D + A \cdot \overline{B} \cdot C \cdot D$$

Como productos de sumas:

$$F(A,B,C,D) = D \cdot (A+B) \cdot (B+C) \cdot (\overline{A}+\overline{B}+\overline{C})$$

3. Se quiere implementar un sistema con dos luces de alarma (diodos LED) y tres sensores (entradas digitales). Llamaremos A y B a las luces de alarma, y x_2, x_1 y x_0 a los sensores digitales. El sistema deberá funcionar de la siguiente manera:

* La alarma A se dispara si se recibe señal del sensor x_2 exclusivamente.
* La alarma B se dispara si se recibe señal del sensor x_0 exclusivamente.
* Las dos alarmas se disparan si se recibe señal de al menos dos sensores cualesquiera.

a) Realizar una especificación tabular del sistema de alarma (tabla de verdad).
b) Realizar una implementación con puertas AND-OR.
c) Realizar una implementación con puertas NAND.
d) Realizar una implementación con puertas NOR.

a)

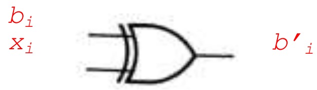
x_2	x_1	x_0	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

b) $A = x_2 + x_1 \cdot x_0$ c) $A = \overline{\overline{x_2} \cdot \overline{x_1 \cdot x_0}}$ d) $A = \overline{\overline{(x_0 + x_2)} + \overline{(x_1 + x_2)}}$

$B = x_0 + x_1 \cdot x_2$ $B = \overline{\overline{x_0} \cdot \overline{x_1 \cdot x_2}}$ $B = \overline{\overline{(x_0 + x_2)} + \overline{(x_0 + x_1)}}$

4. Se desea diseñar un circuito combinacional que realice el complemento a 2 de un número binario de 4 bits. En el diseño se emplearán puertas OR y XOR. Las salidas de las puertas XOR serán las salidas del circuito.

SALIDA DEL CIRCUITO



b_3	b_2	b_1	b_0	b'_3	b'_2	b'_1	b'_0	x_3	x_2	x_1	x_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	0
0	0	1	0	1	1	1	0	1	1	0	0
0	0	1	1	1	1	0	1	1	1	1	0
0	1	0	0	1	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1	1	1	1	0
0	1	1	0	1	0	1	0	1	1	0	0
0	1	1	1	1	0	0	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	1	1	1	1	1	0
1	0	1	0	0	1	1	0	1	1	0	0
1	0	1	1	0	1	0	1	1	1	1	0
1	1	0	0	0	1	0	0	1	0	0	0
1	1	0	1	0	0	1	1	1	1	1	0
1	1	1	0	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	1	1	1	1	0

$b'_i = b_i \oplus x_i$

$x_0 = 0$ $x_1 = b_0$ $x_2 = b_1 + b_0$ $x_3 = b_2 + b_1 + b_0 = b_2 + x_2$

5. Diseñar un circuito combinacional cuya entrada sea un número menor o igual que 15 y cuya salida sea la parte entera de su raíz cuadrada debidamente codificada. Dicho circuito debe tener también una línea de salida que indique si el número introducido era o no cuadrado perfecto.

A	B	C	D	N_1	N_0	P
0	0	0	0	0	0	1
0	0	0	1	0	1	1
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	0

N_1

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$N_1 = A + B$$

N_0

AB \ CD	00	01	11	10
00	0	1	1	1
01	0	0	0	0
11	1	1	1	1
10	0	1	1	1

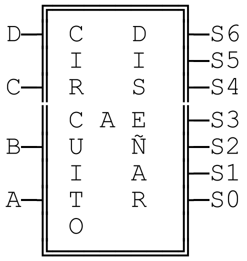
$N_0 = AB + \overline{B}C + \overline{B}D$ o bien $N_0 = (A+\overline{B})\cdot(B+C+D)$

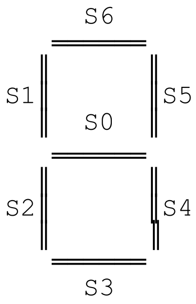
P

AB \ CD	00	01	11	10
00	1	1	0	0
01	1	0	0	0
11	0	0	0	0
10	0	1	0	0

$P = \overline{A}\cdot\overline{C}\cdot\overline{D} + \overline{B}\cdot\overline{C}\cdot D$ o bien $P = \overline{C}\cdot(\overline{B}+\overline{D})\cdot(\overline{A}+D)$

6. Diseñar un circuito al que se le introducen los 4 bits de un código hexadecimal y cuya salida es la excitación para activar un display de 7 segmentos, de acuerdo con la figura adjunta. Los caracteres hexadecimales que no son numéricos, deben aparecer en mayúsculas, excepto la "**b**" y la "**d**", que deben aparecer en minúscula. **NOTA:** Tomar D como el bit más significativo.





DISPLAY 7 SEGMENTOS

a) Escribir la tabla de verdad de todas las funciones booleanas que aparecen.
b) Implementar las funciones de la siguiente forma:

- b1) S1 con puertas AND/OR en dos niveles.
- b2) S2 con puertas OR/AND en dos niveles.
- b3) S4 usando sólo puertas NOR.
- b4) S5 usando sólo puertas NAND.

Hexa	D	C	B	A	S_6	S_5	S_4	S_3	S_2	S_1	S_0
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	0	1	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
b	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
d	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

$$S_1 = \overline{A}\cdot\overline{B} + C\cdot\overline{D} + \overline{C}\cdot D + B\cdot D$$

$$S_2 = (\overline{A} + \overline{B} + D)\cdot(A + B + C)\cdot(A + B + \overline{C} + D)$$

$$S_4 = \overline{\overline{(\overline{B} + \overline{C} + \overline{D})} + \overline{(A + \overline{C} + \overline{D})} + \overline{(A + \overline{B} + C + D)}}$$

$$S_5 = \overline{\overline{(\overline{C}\cdot\overline{D})}\cdot\overline{(\overline{A}\cdot\overline{C})}\cdot\overline{(A\cdot\overline{B}\cdot D)}\cdot\overline{(A\cdot B\cdot\overline{D})}}$$